

A DISTRIBUTED COMPONENT-ORIENTED ARCHITECTURE FOR REAL-TIME ECG DATA ACQUISITION SYSTEMS

V.A. Nagin, I.V. Potapov, S. V. Selishchev

Biomedical Systems Department, Moscow State Institute of Electronic Technology, Moscow, Russia

Abstract-The paper presents a distributed component-oriented multi-tier architecture as applied to the multifunctional ECG software system development. This approach makes it possible to design flexible and manageable ECG applications with the functionality that can be dynamically changed or extended. As a practical implementation of the elaborated architecture a real-time ECG system was developed and its performance parameters were investigated. A set of implemented components includes several real-time data acquisition modules, fast visualization and printing components as well as noise suppression and QRS detection and survey modules. The modular structure of the application with support of transparent integration of new components enabled us to develop additional add-on modules for carrying out specific experiments with ECG transmission methods.

Keywords - ECG, electrocardiography, COM+, client/server, multi-tier architecture.

I. INTRODUCTION

Modern software ECG applications are complex systems consisting of a quantity of different components carrying out different diagnostic and service functions [1]. Common approach to development of these systems is creation of a monolithic application with predefined functionality provided by a fixed structure set of components. Being programmed to perform a fixed number of functions such a system is only able to carry out tasks that were defined and detailed at the stage of design and implementation. Facts named above dictate a number of requirements for modern ECG acquisition and processing systems, which can be grouped into two categories.

Architectural requirements:

- 1) Modular structure;
- 2) Flexible interoperability model between components allowing cross-network and cross-platform communication;
- 3) Open architecture for seamless integration of new components;

Functional requirements:

- 1) Easy to learn user interface;
- 2) Wide application sphere (clinical, laboratory and scientific use);
- 3) Ability to support acquisition devices from different manufactures;
- 4) Transparent integration into medical information systems.

The aim of this study is development and effectiveness investigation of a rational architecture that can meet both functional and architectural requirements best. As a practical implementation of the presented software model a laboratory

real-time ECG system with a number of various components was designed and investigated.

II. DESIGN CONCEPTS

A comparative study of the most widespread software component client/server platforms including Component Object Model (COM+)[2], JavaBean/Remote Method Invocation (RMI) and Common Object Request Broker Architecture (CORBA)[3] showed slight advantage of the first one for real-time applications. Basic concepts of the selected component platform led to the following design principles for the system architecture:

- 1) Microsoft component model COM+ conformance;
- 2) Multi-tier structure;
- 3) Unified data management;
- 4) Open interface implementation for seamless integration of third party components.

COM+ conformance implies use of MS Windows NT/Windows 2000 operating systems or Windows 98/Me for the reduced functionality. System module structure is built on the “document-view” principles according to distributed three-tier (data, business, presentation) design concepts also known as Windows DNA – Distributed interNet Applications Architecture (Fig. 1).

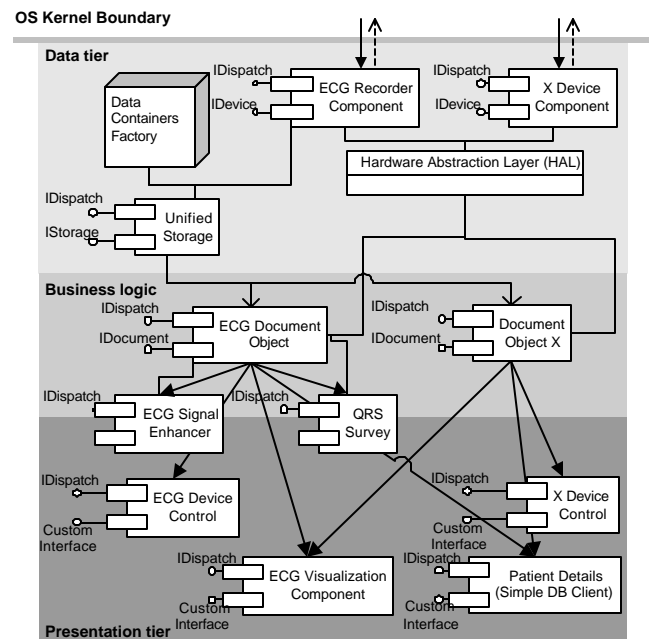


Fig. 1. Multi-tier distributed architecture of the ECG application.

Report Documentation Page

Report Date 25 Oct 2001	Report Type N/A	Dates Covered (from... to) -
Title and Subtitle A Distributed Component-Oriented Architecture For Real-Time ECG Data Acquisition Systems		Contract Number
		Grant Number
		Program Element Number
Author(s)	Project Number	
	Task Number	
	Work Unit Number	
Performing Organization Name(s) and Address(es) Biomedical Systems Department Moscow State Institute of Electronic Technology Moscow, Russia		Performing Organization Report Number
Sponsoring/Monitoring Agency Name(s) and Address(es) US Army Research, Development & Standardization Group (UK) PSC 802 Box 15 FPO AE 09499-1500		Sponsor/Monitor's Acronym(s)
		Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited		
Supplementary Notes Papers from 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, October 25-28, 2001, held in Istanbul, Turkey. See also ADM001351 for entire conference on cd-rom., The original document contains color images.		
Abstract		
Subject Terms		
Report Classification unclassified	Classification of this page unclassified	
Classification of Abstract unclassified	Limitation of Abstract UU	
Number of Pages 3		

The data logic tier, which can be also addressed as data source layer, consists of components responsible for I/O operations with external data acquisition devices such as, for instance, an ECG recorder or a blood pressure monitor. Moreover, the unified data storage component with an object factory capable of instantiating different data carriers also belongs to the data tier. Business logic is implemented by so-called “document” objects, which contain methods for data handling and processing. “View” objects implement some business logic functionality in cases when they are rigidly dependent from the special visualization features or user input. However, the main task of the “view” objects is implementation of the presentation logic, i.e. providing reaction to the user input and proper visualization of the data processed by the business tier. It is worthy of note that objects of business and data tier are not bound to one hosting system. On the contrary, they can be distributed over the network owing to services automatically provided by COM+.

III. IMPLEMENTATION

A. COM+ infrastructure for ECG applications

Microsoft Visual C++ 6.0 compiler with a standard set of software packages including MFC (Microsoft Framework Classes), ATL (Active Template Library) and C++ Standard Template Library (STL) was used to design all system components. Several supporting libraries implementing base system functionality and corresponding COM-component templates were developed according to the elaborated system interface specifications. Developed COM-oriented object libraries provide unified data access and treatment, basic interface implementation and component infrastructure for dynamic object instantiation and interoperability.

B. COM+ oriented ECG acquisition system IntelliCard

A laboratory ECG diagnostic system IntelliCard was developed on top COM+ infrastructure including several components, which can be used as building blocks for any real-time ECG system. The structure of the developed software package can be seen on Figure 1.

Data and business tier components implemented as pure COM components with support of automation interfaces; presentation tier contains fully functional ActiveX components supporting common manipulation interfaces [4]. The software provides measurement and analysis features of:

1. Standard 12-channel ECG;
2. 6-channel ECG;
3. 3-channel ECG.

It is not necessary to switch between these modes; the hardware automatically detects which electrodes are applied. A PC or laptop system with the following features is required to run IntelliCard software for ECG acquisition and automated diagnosis:

1. Operating system Windows 95 with Internet Explorer 4.0 (or later), Windows 98 or Windows NT 4.0 (or later);
2. 20 Mbytes of free hard-disk space;
3. CD-ROM drive;
4. 32 Mbytes of RAM.

The ECG is measured with 500Hz sampling rate and 22bit Analog-to-Digital conversion [5] what guaranties a very high recording accuracy for further analysis [6]. The interface part of the ECG acquisition device is implemented as PCMCIA card; therefore measurements are possible both with desktop PC systems and mobile computers (Fig. 2).

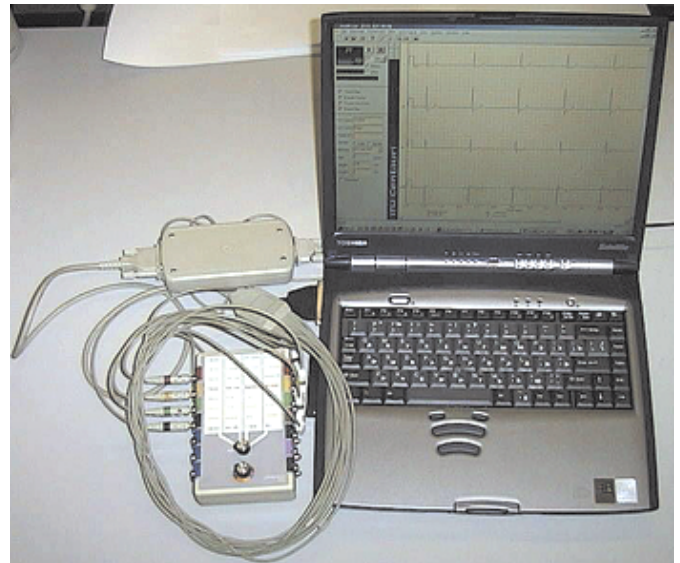


Fig. 2. IntelliCard ECG acquisition and processing system. The system consists of ECG amplifier, interface PCMCIA board, notebook computer and optional ECG signal generator.

As an addition to acquisition mode software supports evaluation mode, which can be used to browse the identified characteristic amplitudes and intervals of the detected QRS complexes. It is possible to look through them and to compare the identified amplitudes and intervals to the norms according to the patient age and gender.

C. Architecture performance tests

Several performance parameters of the entire system were evaluated during numerous tests, which were performed on a PC computer with following characteristics: AMD Athlon 800 Mhz, RAM 128 Mb, Windows 2000.

The ECG application was prepared for instrumentation and executed several times in the real-time acquisition mode, which has the greatest computational burden, with an attached profiler to collect necessary statistic information.

Evaluated parameters, which include overall distribution of CPU utilization time between system components, are presented in the Table 1. The absolute values of the CPU time spent in the particular routine can hardly characterize overall

system performance. However, the ratio of the time spent in COM data handling routines (unified data management) to the total time allocated for the whole application is an important indicator of COM infrastructure effectiveness. It defines the additional computational burden caused by implementing component infrastructure.

TABLE I
TOTAL SYSTEM PERFORMANCE

System component	Distribution of CPU utilization time (percents)		
	Total time (%)	Time spent in COM handling procedures (%)	Time spent in COM handling routines relative to total time(%)
ECG recorder	5.0%	0.05%	1%
Data container factory	0.1%	0.1%	100%
Storage component	0.2%	0.05%	25%
ECG document	1.5 %	0.8%	53%
ECG signal enhancer	3.0%	0.3%	10%
ECG visualization	14%	0.08%	<1%
Patient DB client	0.2%	0.0	0
ECG device control	0.2%	0.08%	40%
Other ECG system modules	1%	0	0
Total ECG system time consumption without system idle time	<26%	<1.5%	<6%

IV. EXPERIMENTAL RESULTS

A. Wireless ECG data transmission

A special ECG transmission add-on module was designed by request of GlobalTel Ltd. (Russian GlobalStar satellite system subsidiary), which launches data transmission service with speed 9600 bps, and plans to improve it to 57600 bps soon. This COM component makes it possible to perform series of ECG transmission tests via satellite phone. Obtained results let us conclude that it is possible to transmit at the real time one channel of ECG for the 9600 bps and up to 6 channels - for 57600 bps. Transmission of 15 seconds of standard 12 lead compressed ECG can be done within 1 minute excluding connection procedures.

B. Evaluation of ECG device interfaces

Evolution of ECG analysis algorithms requires higher sampling rates and analog-to-digital conversion resolution, increase of electrodes number. We had to evaluate during development process different standard computer interfaces, which are able to provide enough data throughput in the real-time environment. Using extensibility features of our ECG software several testing device modules were developed for support of SCSI, USB and Ethernet interfaces. These interfaces were used to connect 2 PC desktops, one of which emulated the activity of the cardiograph hardware sending prerecorded ECG via the interface and the other one was equipped with the ECG software performing real-time

acquisition. During the ECG capture process using different interfaces we evaluated the reliability and utilization of a computer main processor. Those parameters in conjunction with information about difficulty of hardware implementation can be useful while planning the development of new ECG system (see Table II).

TABLE II
PERFORMANCE AND FUNCTIONAL CHARACTERISTICS OF COMMON HIGH-SPEED COMPUTER INTERFACES

Interface	Results		
	CPU resources utilization (%)	Real-time stability (points) (5-best, 1-worst)	Difficulty of implementation of the hardware
SCSI	5%	5	Relatively simple
USB	20%	3	Simple
Ethernet	10%	4	Difficult

V. CONCLUSION

The COM+ technology appears to be a very effective tool for building multifunctional distributed ECG applications. Costs of supporting COM architecture in terms of the CPU utilization time and the additional source code overhead are relatively low. Provided that a basic system infrastructure is implemented and tested development and integration of new components can be very easy. On the other hand, component development reduces the developer freedom and enforces to obey the common protocols.

ACKNOWLEDGMENT

Authors warmly thank colleagues from IfU GmbH, a company headquartered in Chemnitz, Germany, for their participation in the work and carrying out "field tests". Special thanks are directed to Mr. Haase, who did a great deal of routine work with intents to improve the ECG system quality.

REFERENCES

- [1] Association for the Advancement of Medical Instrumentation. American national standard: ambulatory electrocardiographs. Arlington, VA: ANSI/AAMI; 1994. EC 38.
- [2] Distributed Component Object Model Protocol Version 1.0 Microsoft, January 1998.
- [3] CORBA/IIOP 2.2 Specification Version 2.2 June 1998.
- [4] Nagin V.A., Potapov I.V., Selishchev S.V., Sharf V. A Software Package for Acquisition and Mathematical Processing of ECG Data on the Basis of COM Architecture. Meditsinskaya Tekhnika, vol. 1, Moscow-2001. -P.3-7. // English translation is available in Biomedical Engineering journal published by Kluwer Academic/Plenum Publishers.
- [5] Kurekov S. F., Prilutski D. A., Selishchev S. V. Sigma-Delta Analog-to-Digital converters for Biomedical Data Acquisition Systems // Proceeding of 4th European conference on engineering and medicine.- Warsaw, May 25-28, 1997.-P.163-164.
- [6] Cohen A. Biomedical Signals: Origin and Dynamic Characteristic; Frequency-Domain Analysis. // In The Biomedical Engineering Handbook, Editor-in-Chief J.D.Bronzino // CRC and IEEE Press. -Boca Raton, Florida.-1995. -P. 805-827.